

# Referat Laboratorul 1 - Superscalare Avansata

---

Deaconu Ioan  
April 24, 2015

## 1 INTRODUCERE

Încă de la primele generații de procesoare create vreodată s-a dorit îmbunătățirea vitezei de calcul. Primele procesoare funcționau secvențial, astfel o singură instrucțiune era executată la un moment dat.

Pentru a putea mări viteza de calcul a procesoarelor, acestea au implementat executarea unei instrucțiuni sub forma unui pipeline. Această metodă a împărțit executarea unei instrucțiuni în mai multe stadii, astfel toate stagiile sunt folosite de mai multe instrucțiuni în același timp. În acest document vom analiza evoluția procesoarelor Intel superscalare.

## 2 NOTIUNI TEORETICE AFERENTE

Pentru a putea executa o instrucțiune, aceasta trebuie citită, decodată, executată și scrise datele înapoi în memorie. Acești pași au fost împărțiți în componente simple, pentru a putea face procesorul mai modular. Acești pași, în arhitectura clasică RISC sunt:

- Instruction Fetch.
- Instruction Decode and register Fetch.
- Execute.
- Memory access.
- Write Back.

## 2.1 SCALAR - PIPELINE

Din cauza faptului ca doar un singur modul este activ iar celelalte nu fac nimic, incepand cu ani 70 s-a folosit principiul de pipeline. Astfel, pentru a se eficientiza utilizarea procesorului, mai multe instructiuni vor fi executate in acelasi timp, dar decalate cu un pas, pentru a putea folosi modulele mai eficient.

De exemplu, in cazul a 5 instructiuni, daca consideram ca toate stagiile dureaza 1 unitate de timp, atunci fara pipeline ar dura 25 de unitati de timp executia totala a instructiunilor. Folosind pipeline, timpul total de executie al celor 5 instructiuni ar fi de doar 9 unitati de timp [2].

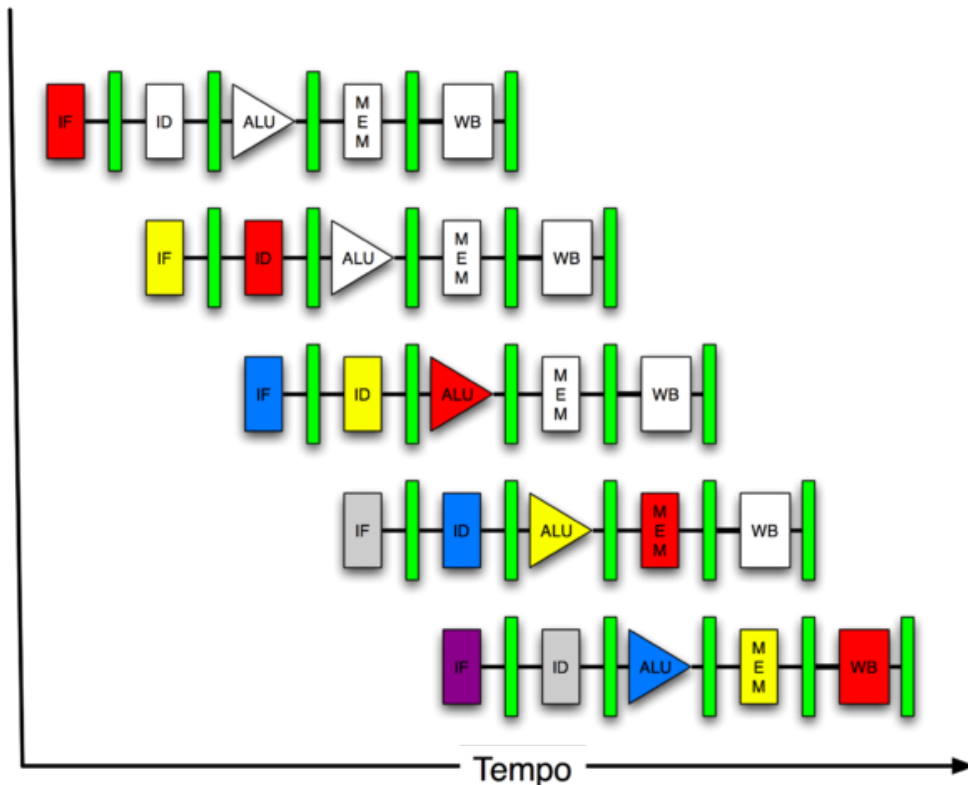


Figure 2.1: Exemplu de executie a instructiunilor folosind pipeline.

## 2.2 SCALAR AVANSAT

Cu timpul, odata ce unitatile interne ale procesorului au devenit din ce in ce mai puternice, pipelineul a devenit din ce in ce mai lung, ajungand in Pentium 4 la un numar de 31 de stagii de pipeline. Desi teoretic un pipe lung permite o granularitate mai buna, dezavantajul major il constituie branchurile, deoarece tot pipeline-ul trebuie golit si incarcat cu instructiuni noi. Deasemenea un pipeline mai lung inseamna o durata mai mare de executie a unei instructiuni,

dar permite atingerea unor frecvente mai mari.

Deoarece exista unitati care pot fi nefolosite, spre exemplu floating point unit nu este folosita cand se executa o instructiune de load sau store, dezvoltatorii de procesoare au decis sa execute mai multe instructiuni in paralel. Astfel pipelineul pe langa numarul de stagii care le are a fost marit si in latime. Mai exact, in loc de executia unei singuri instructiuni, se citesc simultan mai multe instructiuni si se executa simultan. Acest mod de abordare ar permite , de exemplu ca in acelasi timp in care se executa adunarea a 2 registre sa se poata compara alte registre[3].

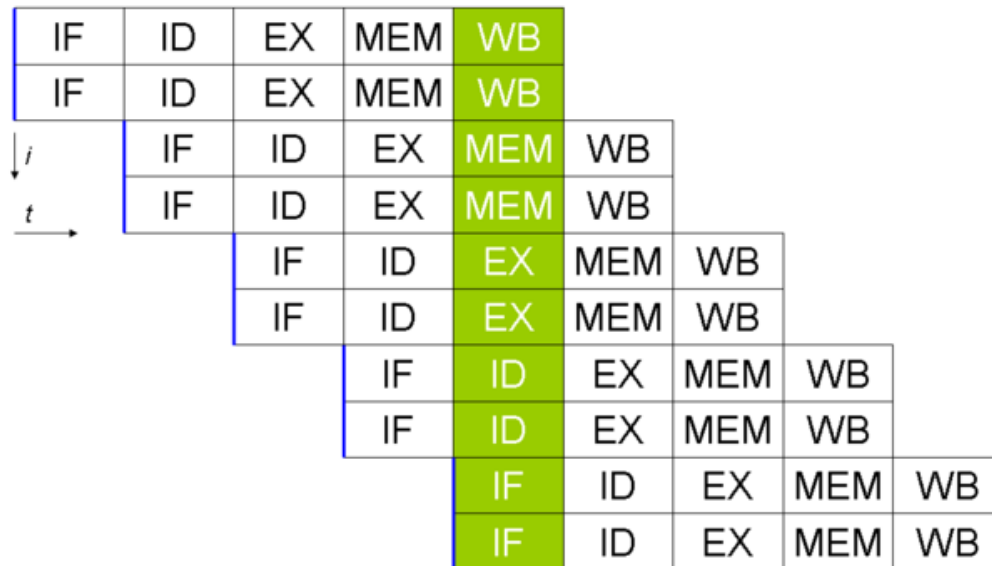


Figure 2.2: Exemplu de executie a mai multor instructiunilor folosind super scalar.

### 3 IMPLEMENTARE

Primul procesor superscalar de 32 de biti creat de intel a fost intel Pentium, in 1993, bazat pe microarhitectura P5. Acesta are 2 pipeline-uri care permit executarea a 2 instructiuni in paralel.

Primul pipeline (U) este pipeline-ul principal, in care se poate executa orice instructiune citita. Cel de-al doilea pipeline (V) este un pipeline mai simplu, care poate executa majoritatea instructiunilor, cu putine exceptii[4].

Numarul de stagii in acest pipeline este de 5 [1], dar cu fiecare arhitectura nou creata, numarul de stagii in lungime a crescut, astfel primele procesoare pentium 4 avea un pipeline de 20 de stagii, comparativ cu cele 10 stagii ale lui pentium 3. Acest lucru a fost necesar pentru a putea marii frecventa procesoarelor. Desi mai multe stagii de pipeline inseamna un timp mai mare pentru executarea unei instructiuni, inseamna si stagii mai simple, cu mai putini tranzistori care pot rula la frecvente mai mari. Din aceasta motiv in cazul lui pentium 4 Prescott, numarul de stagii a ajuns la un numar de 31.

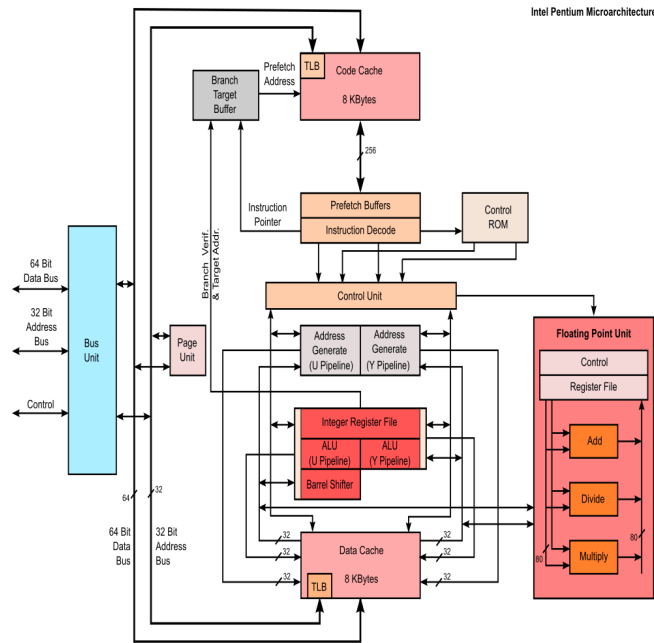


Figure 3.1: Microarhitectura P5



Figure 3.2: Cele 20 de stagii ale lui pentium 4

Generatia urmatoare, microarhitectura core, a micșorat numarul de stagii la 12-14, ceea ce a dus la reducerea frecvenței de lucru, dar pentru a compensa reducerea frecvenței s-a dublat latimea pipeline-ului, de la 2 instructiuni executate in paralel, la 4 instructiuni executate in paralel. Consumul a fost mult micșorat, scazand de la 130 Watti la 65 de Watti, in principal datorita frecvenței reduse, dar performanta acestora este net superioara, chiar daca functioneaza la frecvențe reduse.

Urmatoarea arhitectura, intel Core i7(Nehalem) a marit iar numarul de stagii de pipeline la 20 (daca operatia de cache a fost de tip hit) sau 24 (daca operatia a fost de tip miss) ca in generatiile urmatoarea, incepand de la Sandy Bridge, si continuand cu Ivy Bridge si Haswell sa foloseasca un pipeline cu 14 stagii pentru cache hit si 19 stagii pentru cache miss. Numarul micșorat de stagii fata de Nehalem au permis o imbunatatire a IPC-ului de pana la 20%.

### 3.1 SMT

Arhitectura superscalara poate executa doua sau mai multe instructiuni in paralel, dar doar instructiuni ale aceluiasi program. Din moment ce pipeline-ul a fost paralelizat, urmatorul pas a fost simularea unui nou nucleu. Acest lucru a fost posibil dubland registrii de baza, pentru a

putea avea 2 program counter si a putea rula instructiuni de la adrese diferite, instructiuni care nu erau consecutive. Chiar daca sistemul de operare vedea 2 nuclee fizice, in realitate era un singur nucleu fizic, care putea rula instructiuni de la 2 programe diferite in paralel. Acest lucru permite folosirea unitatilor procesorului la capacitate maxima.

## 4 SIMULARE

Simultaneous Multi-Threading reprezinta cea mai avansata forma de arhitectura superscalara. Pentru a verifica avantajele ei, am inclus un benchmark din articolul [5].

In acel test, se verifica cat de mult poate mari viteza de executie folosind SMT. Dupa cum se observa pe rezultate, speedup-ul mediu este de aproximativ 1.2. Exista cazuri in care speedup-ul este sub 1, dar sunt rare. Acest test arata ca prin dublarea registrilor si folosirea arhitecturii superscalare deja existente, o marirea de aproximativ 5% a unui nucleu rezulta intr-o crestere a performantei de pana la 40%.

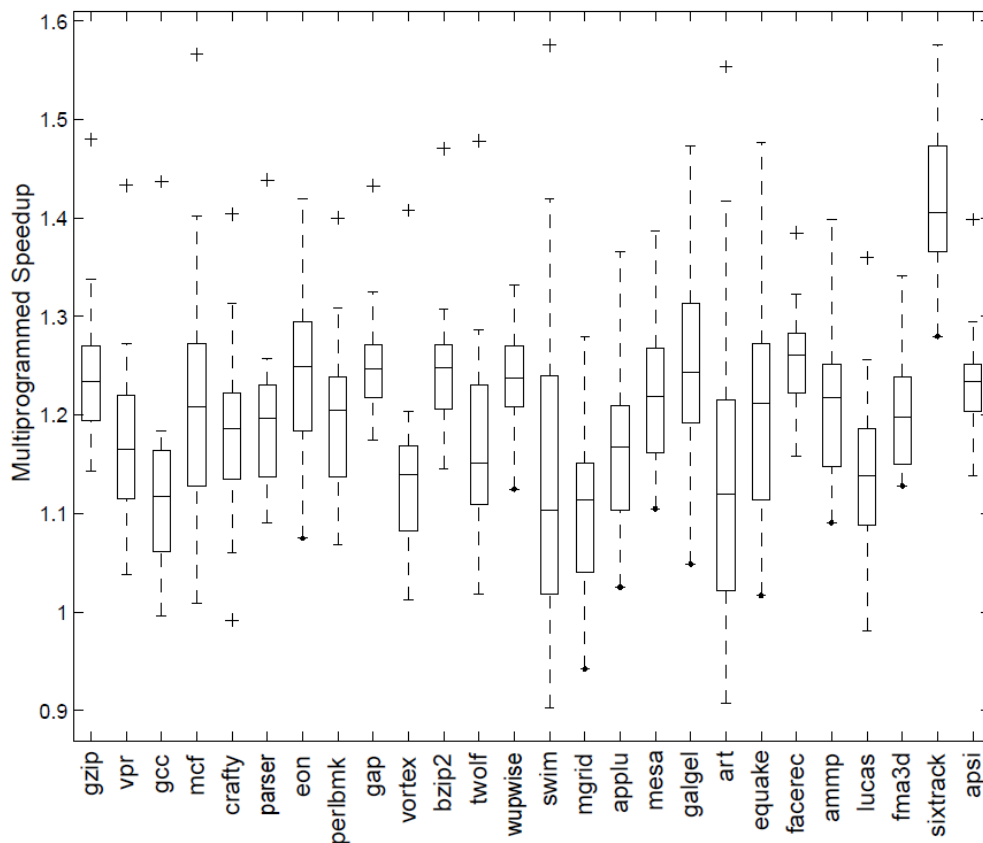


Figure 4.1: Speed-up obtinut folosit SMT

## 5 CONCLUZII

Superscalarea este intalnita in toate microprocesoarele de la ora actuala. Fara ea, performantele actuale nu ar fi posibile , cel putin nu cu un consum mic si un pret accesibil. Aceasta tehnica se dezvolta mereu, existand deja multe implementari diferite ale acesteia, un exemplu fiind procesoarele AMD pe arhitectura Bulldozer.

## REFERENCES

- [1] W.-M. W. Hwu, S. A. Mahlke, W. Y. Chen, P. P. Chang, N. J. Warter, R. A. Bringmann, R. G. Ouellette, R. E. Hank, T. Kiyohara, G. E. Haab, et al. The superblock: an effective technique for vliw and superscalar compilation. *the Journal of Supercomputing*, 7(1-2):229–248, 1993.
- [2] M. Johnson and M. Johnson. *Superscalar microprocessor design*, volume 77. prentice Hall Englewood Cliffs, New Jersey, 1991.
- [3] N. P. Jouppi and D. W. Wall. *Available instruction-level parallelism for superscalar and superpipelined machines*, volume 17. ACM, 1989.
- [4] S. Palacharla, N. P. Jouppi, and J. E. Smith. *Complexity-effective superscalar processors*, volume 25. ACM, 1997.
- [5] N. Tuck and D. M. Tullsen. Initial observations of the simultaneous multithreading pentium 4 processor. In *Parallel Architectures and Compilation Techniques, 2003. PACT 2003. Proceedings. 12th International Conference on*, pages 26–34. IEEE, 2003.